PROGRAMMING TASK

In this part, you must implement your own graph data structure by taking inspiration from your textbook and use it to help to solve problem. You are not allowed to use any external library or .jar file. Any solutions without using graph data structure are not evaluated!

In this homework, you are expected to implement Trie data structure and following functions.

Firstly, you should create a Trie data structure, and then you should insert all words read from user. You can implement more than one Trie data structure. Finally, you should complete following functions:

Question 1(25 points):

Boolean Search (String arg): This function should return true if given argument is in your Trie, otherwise it should return false.

Void countPrefix(Trie trie): This function takes all the strings in Trie and checks if a string is a prefix of other strings in Trie. It prints the number of occurrences for each string. For this function, you may use a symbol table that keeps track of the number of appearances of each key.

Note: You can change the parameter for this method if you need!

Void reverseFind (String suffix): This function should print all strings end with given suffix in your Trie, lexicographically. For this function, you may consider using multi-trie solution, or research and use more complex data structures such as suffix arrays.

Sample Input:

```
7               #number of inputs
br              #word 1
bridge
brick           ...
dogma
cat
cathedral
dog             #word 7
```

Sample Output-1

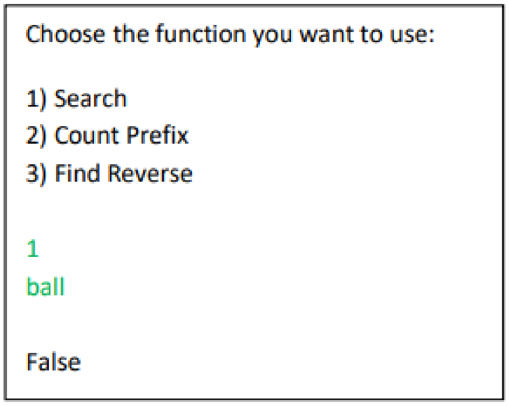
```
Choose the function you want to use:

1) Search
2) Count Prefix
3) Find Reverse

1
cat

True
```

Sample Output-2:



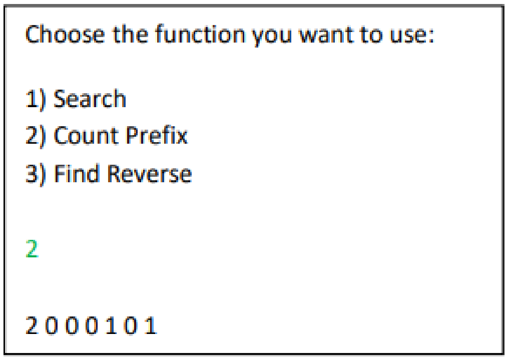
```
Choose the function you want to use:

1) Search
2) Count Prefix
3) Find Reverse


1
ball

False
```

Sample Output-3:



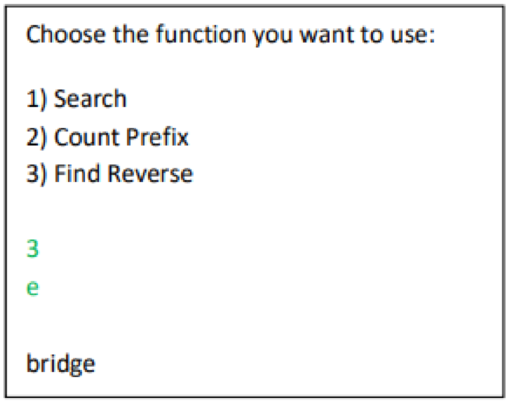
```
Choose the function you want to use:

1) Search
2) Count Prefix
3) Find Reverse


2

2 0 0 0 1 0 1
```

Sample Output-4:



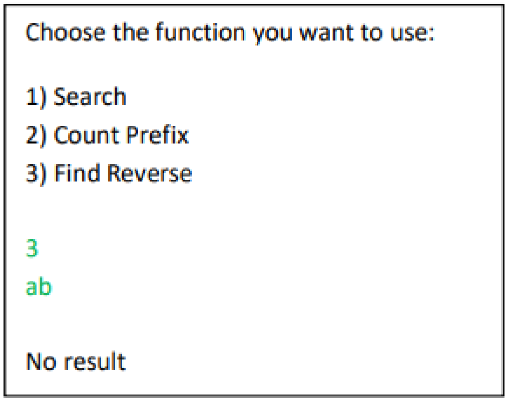
```
Choose the function you want to use:

1) Search
2) Count Prefix
3) Find Reverse


3
e

bridge
```

Sample Output-5



```
Choose the function you want to use:

1) Search
2) Count Prefix
3) Find Reverse


3
ab

No result
```

Question 2(25 points):

In this question, you will implement a String sorting algorithm. You will be given two same length of arrays of String. The Strings of the second array will not contain duplicate characters. You will sort the first array based on the following rules.

First, you need to find the distance between the Strings at the same index of the arrays. The distance calculation formula is:

- Each letter has a number starting from 1 and increasing one by one. “A” is 1, while “Z” is 26.
- The integer value of a String is determined by the letter number. For an example, the integer value of “abj” is 1210.

- Find the distance between two Strings at the same indexes by subtracting their values. For example, the distance between at index “0” from the first array “abj” and from the second array “bal” is |1210-2112| = 902.
- If the distance value is even, then sort your String from the first set based on the String from the second set. In our example “abj” will be sorted based on the order of “bal”, which ‘b’ has the highest priority while “l” has the least in sorting. The sorted version of “abj” will become “baj”. The order of letter “j” in the first string cannot be found in the second string. Therefore, you need to add it to the last of the String.
- If the distance value is odd, then sort your string lexicographically in ascending order.

Example-1

First Array:
dog honey apple rope

Second Array:
gdbo bonex pina elo

Sorted Array:
gdo ehnoy ppale eorp

Example-2

First Array:
forest water nick doze

Second Array:
ftrki olis cim ipgk

Sorted Array:
eforst aertw cink doze